5Г

УДК 519.7

И.С. Грунский, С.В. Сапунов
Донецкий национальный технический университет
Украина, 83001, г. Донецк, ул. Артема, 58
Институт прикладной математики и механики НАН Украины, г. Донецк
Украина, 83114, г. Донецк, ул. Розы Люксембург, 74

Восстановление графа операционной среды мобильного робота путем разметки вершин, пригодной для дальнейшей навигации

I.S. Grunsky, S.V. Sapunov
Donetsk National Technical University
Ukraine, 83001, c. Donetsk, Artema st., 58
Institute of Applied Mathematics and mechanics of NAS of Ukraine, Donetsk
Ukraine, 83114, c. Donetsk, Rozy Luxemburg st., 74

Reconstruction of the Graph of Operating Environment of Mobile Robot by Vertex Labeling Sufficient for Further Navigation

І.С. Грунський, С.В. Сапунов
Донецький національний технічний університет
Україна, 83001, м. Донецьк, вул. Артема 58
Інститут прикладної математики і механіки НАН України, м. Донецьк
Україна, 83114, м. Донецьк, вул. Рози Люксембург, 74

Відновлення графа операційного середовища мобільного робота шляхом позначення його вершин, яке сприяє подальшій навігації

Рассматривается задача построения автономным мобильным роботом топологической модели своей операционной среды. Модель среды представляет собой связный неориентированный граф с помеченными вершинами. В работе предложен полиномиальный алгоритм восстановления и разметки графа среды для коллектива из робота-супервизора и робота-исследователя.
Ключевые слова: мобильный робот, топологическая модель среды, восстановление графа.

The problem of robotic exploration of graph-like operating environment is considered. The environment is defined as simple connected undirected vertex labeled graph. In the article, polynomial time graph reconstruction and vertex labeling algorithm for collective of robot-explorer and robot-supervisor is proposed.
Key Words: mobile robot, topological model of environment, graph reconstruction.

Розглянуто задачу побудови автономним мобільним агентом топологічної моделі свого операційного середовища. Модель середовища є зв’язним неорієнтованим графом з позначеними вершинами. В статті запропоновано поліноміальний алгоритм відновлення і розмітки графа середовища для колективу з робота-супервізора і робота-дослідника.
Ключові слова: мобільний робот, топологічна модель середовища, відновлення графа.

Введение

Актуальными проблемами в области разработки информационного обеспечения мобильных роботов являются проблемы представления операционной среды и разработка методов навигации в этой среде [1]. К моделированию таких сред определилось два подхода: метрический, использующий геометрические свойства среды, и топологический, использующий описания связей между различными областями среды [2]. Топологические модели представляют собой ориентированные или неориентированные графы с размеченными различными способами вершинами и/или дугами. Такой подход позволяет применять для решения конкретных задач навигации широкий набор эффективных алгоритмов теории графов [3].

В данной работе под навигацией понимается перемещение мобильного агента (робота, поисковой программы и т.п.) из текущей области его операционной среды в заданную ее область. Полагаем, что в решении задач навигации участвует коллектив, состоящий из агента-супервизора (АС) и транспортного агента (АТ). АС вырабатывает описание пути из текущего местоположения АТ в область назначения и передает его АТ. Используя это описание, АТ самостоятельно проходит путь в среде.

При планировании навигации возникают три взаимосвязанные фундаментальные задачи: задача построения модели неизвестной среды (exploration), задача определения положения робота в известной среде (robot self-location) и задача проверки соответствия неизвестной среды и ее модели (карты) (map validation) [4]. Заметим, что эти задачи имеют соответствующие аналоги в теории автоматов [5].

В данной работе рассматривается построение коллективом агентов модели неизвестной им среды. Коллектив состоит из агента-супервизора (АС) и агента-исследователя (АИ). АИ перемещается по среде и воспринимает локальную информацию, достаточную для сопоставления областям среды элементов модели. Если воспринимаемой информации не достаточно для такого сопоставления, то АИ изменяет среду путем установки искусственных ориентиров. АИ передает информацию о среде и произведенных им ее изменениях АС, который и строить модель среды. По этой модели АС создает описания путей для АТ.

В качестве модели рабочей среды мобильных агентов мы будем рассматривать конечные простые связные неориентированные графы, вершины которых могут быть помечены метками из некоторого множества меток (т.н. помеченные графы). Такая модель возникает, например, при качественной навигации мобильных роботов [6], [7].

Целью данной работы является разработка алгоритма построения коллективом из АС и АИ топологической модели неизвестной ему среды в условиях, когда естественные (т.е. являющиеся частью среды) ориентиры недоступны АИ. В результате выполнения алгоритма в областях среды устанавливаются искусственные ориентиры так, что АТ может осуществлять навигацию по путям, описываемым цепочками этих ориентиров.

Постановка задачи

АИ установлен в произвольную вершину неизвестного ему связного простого неорграфа. Все вершины графа не помечены (или, что равносильно, помечены одной и той же меткой). АИ наблюдает 3-окрестность текущей вершины (т.е. все вершины, удаленные от текущей на расстояние не более чем в 3 вершины). Он может перемещаться по ребрам между смежными вершинами. АИ также может метить текущую вершину меткой (искусственным ориентиром) из некоторого множества меток. При

этом он не может менять метку, которую сам установил. АИ передает АС разметку наблюдаемой им локальной окрестности и имя установленной им метки. АС передает АИ последовательность меток вершин, описывающую путь по графу.

Требуется разработать алгоритм перемещений АИ по графу и разметки пройденных вершин, который позволяет восстановить граф. Причем, полученная разметка должна позволить АТ, наблюдающему 2-окрестность текущей вершины, осуществлять навигацию по графу.

К настоящему времени известно довольно много методов восстановления графов [8]. Однако эти методы, как правило, предполагают разметку всех вершин графа одной и той же меткой, что неприемлемо для последующей навигации. Тривиальным методом решения поставленной задачи является восстановление графа с разметкой каждой его вершины уникальной меткой. В работе предложен алгоритм, в котором на основе анализа структуры графа уменьшается число различных меток.

Основные определения

Неопределяемые понятия общеизвестны и их можно найти в [9].

Помеченным графом назовем конечный простой связный неориентированный граф с помеченными вершинами $G=(V,E,M,\mu)$, где V – множество вершин, $|V|=n$, E – множество ребер (т.е. неупорядоченных пар вершин), M – множество меток, $\mu: V\to M$ – сюръективная функция разметки вершин. Путем в графе G назовем последовательность вершин $p=g_1\dots g_k$ такую, что $(g_i,g_{i+1})\in E$, $i=1,\dots,k-1$. Число $k\in N$ назовем длиной пути p. Кратчайший путь из вершины g в вершину h называется расстоянием между этими вершинами. Меткой $\mu(p)$ пути p назовем слово $w=\mu(g_1)\dots\mu(g_k)$ в алфавите меток M. Будем говорить, что слово w определяется вершиной g_1. Длину слова w будем обозначать через $d(w)$. Начальный отрезок или префикс длины l слова w будем обозначать через $\mathrm{pre}_l(w)$. Конечный отрезок или суффикс длины l слова w будем обозначать через $\mathrm{suf}_l(w)$. Инверсией слова $w=$ $=\mu(g_1)\dots\mu(g_k)$ назовем слово $w^{-1}=\mu(g_k)\dots\mu(g_1)$. Множество L_g всех слов $w\in M^+$, определяемых вершиной $g\in V$, будем называть языком этой вершины. Граф G будем называть приведенным, если для любых вершин $g,h\in V$ из $g\neq h$ следует $L_g\neq L_h$.

Определим на M^+ частичную операцию $\circ$ композиции слов. Пусть $a,b\in M$, $w,v\in M^*$, тогда $wa\circ av=wav$ и $wa\circ bv$ не определено, если $a\neq b$.

Введем операцию $*: V\times M^+\to 2^V$ соотношением: для любой вершины $g\in V$ и любого слова $w\in M^+$ через $g*w$ обозначим множество всех вершин $h\in V$ таких, что существует путь p, соединяющий вершины g и h, и $\mu(p)=w$. Ясно, что если слово $w\in L_g$, то $|g*w|>0$ и $|g*w|=0$ в противном случае.

Под *k*-окрестностью $\Gamma_g^{(2)}$ вершины $g\in V$ будем понимать множество всех вершин, находящихся от g на расстоянии не превосходящем $k\in N$. Число k назовем размером окрестности $\Gamma_g^{(2)}$.Таким образом, имеем:

$$\Gamma_g^{(1)}=\{g\},\ \Gamma_g^{(2)}=\Gamma_g^{(1)}\cup\{h\mid(g,h)\in E\},\ \Gamma_g^{(3)}=\Gamma_g^{(2)}\cup\{s\mid\exists h((g,h)\in E\wedge(h,s)\in E)\}$$

и т.д.

АИ, находясь в вершине $g \in V$, наблюдает метки всех вершин из $\Gamma_g^{(k)}$ для некоторого $k \in N$. Основываясь на анализе «увиденного», агент принимает решение о перемещении в одну из смежных вершин. Будем считать, что для перемещения из вершины g в смежную ей вершину h АИ должен наблюдать h и отличать ее от всех вершин из $\Gamma_g^{(k)}$. Тогда для перемещений по графу АИ должен отличать вершины из 2-окрестности текущей вершины от других наблюдаемых им вершин. АИ может либо произвольно блуждать по графу, либо следовать по заданному пути из текущей вершины. АИ имеет следующие возможности по временному или постоянному изменению разметки вершин. Он может пометить непомеченную вершину меткой из множества M. Он также может устанавливать и подбирать в текущей вершине переносные маркеры (т.н. камни).

В качестве меры сложности агента будем рассматривать размер наблюдаемой им окрестности текущей вершины. Через А*k* обозначим агента, который наблюдает k-окрестность текущей вершины. Заметим, что А2 является простейшим агентом, так как агент А1 не может осуществлять целенаправленное перемещение.

Детерминированная разметка вершин

Обоснуем выбор специальной разметки вершин графа, позволяющей агенту А2 осуществлять навигацию.

Функцию разметки $\mu : V \to M$ будем называть детерминированной или Д-разметкой, если для любой вершины $g \in V$ и любых вершин $s, t \in \Gamma_g^{(2)}$ из $s \neq t$ следует $\mu(s) \neq \mu(t)$. Помеченный граф G с детерминированной функцией разметки будем называть детерминированным или Д-графом.

Рассмотрим свойства Д-графов.

Лемма 1. Помеченный граф G является Д-графом тогда и только тогда, когда для любой вершины $g \in V$ и любого слова $w \in M^+$ выполняется

$$|g * w| = \begin{cases} 1, \text{если } w \in L_g, \\ 0, \text{в противном случае.} \end{cases} \quad (1)$$

Доказательство. Действительно, для всех слов длины 1 и 2 равенство (1) очевидно, а любое слово длины больше 2 единственным образом записывается в виде композиции своих двухбуквенных подслов. С другой стороны, из того, что для любой вершины $g \in V$ и любого слова $w \in M^+$ выполняется $|g * w| \leq 1$ следует, что в $\Gamma_g^{(2)}$ нет одинаково помеченных вершин. Лемма доказана.

Таким образом, Д-граф определяется или через локальные свойства его вершин, или через нелокальное свойство множества всех путей, определяемых каждой из вершин.

Лемма 2. Для любых различных вершин $g, h \in V$ и любого слова $w \in L_g \cap L_h$ расстояние между вершинами $g * w$ и $h * w$ не меньше 4.

Доказательство. Пусть $w = x_1 \dots x_k$, где $x_i \in M$, $1 \leq i \leq k$. Предположим, что расстояние между вершинами $s = g * w$ или $t = h * w$ меньше 4. Пусть это расстояние равно 1, т.е. $s = g * x_1 \dots x_k = h * x_1 \dots x_k = t$ Тогда в окрестности $\Gamma_s^{(2)}$ окажутся две различные вершины с одной и той же меткой x_{k-1}, что невозможно по определению Д-графа.

Следовательно, $g * x_1 \ldots x_{k-1} = h * x_1 \ldots x_{k-1}$. По индукции $g = h$, что невозможно. Пусть расстояние между вершинами s и t равно 2, то есть $(s,t) \in E$. Тогда в окрестности $\Gamma_s^{(2)}$ находится вершина t с меткой $\mu(y) = \mu(s)$, что невозможно по определению Д-графа. Пусть расстояние между вершинами s и t равно 3, то есть существует вершина $q \in V$ такая, что путь sqt является кратчайшим путем из s в t. Тогда $s,t \in \Gamma_q^{(2)}$ и $\mu(s) = \mu(t)$, что невозможно по определению Д-графа. Лемма доказана.

Одной из центральных проблем, возникающих при навигации МА, является проблема самостоятельного определения агентом своего положения в среде (задача самолокализации агента) [2], [4]. В [10], [11] предложено решение задачи самолокализации для приведенных Д-графов. Следующее утверждение показывает, что на вершинах любого конечного связного простого неорграфа может быть построена такая Д-разметка, что полученный в результате Д-граф является приведенным, т.е. для него разрешима задача самолокализации.

Лемма 3. Если в Д-графе G существует вершина с уникальной меткой, то G является приведенным графом.

Доказательство. Будем говорить, что вершины $g, h \in V$ неотличимы, если $L_g = L_h$. Ясно, что отношение неотличимости является эквивалентностью на множестве V. Пусть B, B' – некоторые классы неотличимости вершин и $g \in B$. Пусть $g * w \in B'$. Покажем, что $|B| = |B'|$. Если $h \in B$, то из неотличимости g и h следует, что $h * w \in B'$. По определению Д-графа $h * w \neq g * w$ при $h \neq g$. Поэтому $|B| \leq |B'|$. Обозначим $g' = g * w$ и $h' = h * w$. Из очевидного неравенства $h' * w^{-1} \neq g' * w^{-1}$ следует $|B'| \leq |B|$. Таким образом, если некоторый класс неотличимости одноэлементен, то и все классы неотличимости одноэлементны. Следовательно, Д-граф G является приведенным. Лемма доказана.

Таким образом, если пометить произвольную вершину произвольного Д-графа уникальной меткой, то получим приведенный Д-граф.

Теорема 1. Агент Ak может осуществлять навигацию на Д-графе с $n \geq 3$ тогда и только тогда, когда $k \geq 2$.

Доказательство. Покажем, что агент A2, находящийся в вершине g Д-графа G, получив от АС слово $w \in L_g$, может переместиться из нее в вершину $h = g * w$. Пусть $d(w) = 2$. Тогда, по определению Д-графа, в $\Gamma_g^{(2)}$ находится единственная вершина с меткой $\mathrm{suf}_1(w)$ и этой вершиной является вершина h. A2 «видит», что эта вершина смежная вершине g и может переместиться в нее. Пусть $d(w) > 2$. Тогда слово w единственным образом записывается в виде композиции $w_1 \circ w_2 \circ \ldots \circ w_l$ двухбуквенных подслов w_i, $d(w_i) = 2$, $1 \leq i \leq l$. Из вершины g агент A2 переходит в вершину $g * w \in \Gamma_g^{(2)}$. Из вершины $g * w_i$ он переходит в вершину $g * w_{i+1} \in \Gamma_{g*w_i}^{(2)}$, $1 \leq i \leq l-1$. Наконец, из вершины $g * w_{l-1}$ агент переходит в вершину $g * w_l \in \Gamma_{g*w_{l-1}}^{(2)}$, которая и является вершиной h. Заметим, что агент A1 не может пройти путь $x_1 x_2$ так как, находясь в вершине с меткой x_1, агент не «знает» по какому ребру перейти в вершину с меткой x_2. Теорема доказана.

Таким образом, Д-разметка графа является достаточным условием для того, чтобы простейший агент мог осуществлять навигацию.

Рассмотрим проблему построения Д-разметки на графе. Если G является ациклическим графом, т.е. деревом, то агент А2 может построить на нем Д-разметку, если агент «запоминает» метки пройденных путей. Покажем, что существует граф, который не может быть Д-размечен агентом А2. Рассмотрим пример на рис. 1.

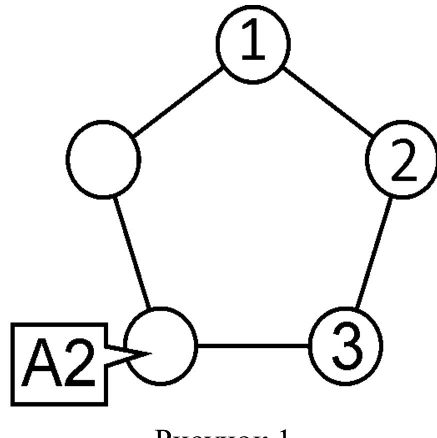


Рисунок 1

Пусть агент А2 уже выполнил Д-разметку трех вершин графа. Перейдя в непомеченную вершину, он наблюдает вершину с меткой 3 и вершину без метки. Предположим, что А2 «запомнил» все установленные им метки. Тогда он не может использовать метки 2 и 3. Так как априори граф не известен А2, то, исходя из доступной ему информации, агент пометит текущую вершину меткой 1 и, тем самым, нарушит детерминированность разметки. Заметим, что агент А3, находясь в той же вершине, «увидел» бы, что метку 1 нельзя использовать для ее разметки. Из леммы 2 следует, что вершины, удаленные друг от друга на расстояние не более чем 3 вершины, не должны иметь одинаковые метки. Таким образом, справедливо следующее утверждение.

Теорема 2. Для Д-разметки произвольного графа агентом Аk достаточно, чтобы k превосходило 2.

Д-разметка вершин графа коллективом агентов

Из теоремы 2 следует, что для построения Д-разметки достаточно использовать агента А3. Построение минимальной Д-разметки на основе только лишь локальной информации о вершинах (т.е без знания всего графа) представляется невозможным в общем случае. Поэтому будем рассматривать «жадный» алгоритм решения этой задачи с помощью А3.

В основу предлагаемого в работе метода восстановления графа положен метод обхода графа в ширину [12]. При этом для каждой вершины $g \in V$ в качестве имени используется метка пути в нее из начальной вершины по дереву обхода (этим объясняется выбор в пользу метода обхода в ширину: так как путь в любую вершину из начальной по дереву обхода имеет кратчайшую длину среди всех таких путей, то и длина имени этой вершины также кратчайшая).

Зададим линейный порядок на множестве M. Без потери общности можно предположить, что $M \subseteq N$ и, что перед началом работы алгоритма все вершины исследуемого графа помечены меткой 0.

Для каждой вершины $g \in V$ хранятся ее имя $name(g)$, ее метка $\mu(g)$, множество $Al(g)$ смежных с ней вершин и множество $Ll(g)$ меток этих вершин. Для хранения множества имен вершин, окрестности которых еще не обработаны, используется список T типа FIFO. Для хранения имен всех вершин используется множество W.

Первоначально АИ устанавливается в неизвестную ему вершину s исследуемого графа G. АИ считывает метки вершин из $\Gamma_s^{(3)}$, корректирует множество доступных для Д-разметки меток, метит s наименьшей из них и передает АС метку $\mu(s)$. Если требуется получить приведенный граф, то метка $\mu(s)$ больше не используется. АС вычисляет имя $name(s)$ (для начальной вершины $name(s) = \mu(s)$) и добавляет его в T и W.

Пусть на i-ой итерации АИ находится в некоторой уже помеченной им вершине $g \in V$. Для всех вершин с меткой 0 из $\Gamma_g^{(2)}$ АИ выбирает любую из них, например t, и перемещается в нее. АИ считывает метки вершин из $\Gamma_t^{(3)}$, корректирует множество доступных для Д-разметки меток, метит t наименьшей из них и передает АС метку $\mu(t)$. АС добавляет $\mu(t)$ в $Ll(g)$, вычисляет имя $name(t)$ как конкатенацию $name(g)\mu(t)$ и добавляет его в T, W и $Al(g)$. Далее, АС добавляет $name(g)$ и $\mu(g)$ в $Al(t)$ и $Ll(t)$ соответственно. После этого АИ возвращается в вершину g.

Окончив разметку $\Gamma_g^{(2)}$, коллектив агентов приступает к проверке ребер, поперечных дереву обхода. АИ считывает метки всех вершин из $\Gamma_g^{(2)}$ и передает АС множество $\mu(\Gamma_g^{(2)})$. Если $\mu(\Gamma_g^{(2)}) \setminus Ll(g) \neq \varnothing$, то АС передает АИ произвольную метку из $\mu(\Gamma_g^{(2)}) \setminus Ll(g)$, например метку $\mu(q)$ вершины q. АИ переходит в q и оставляет в ней камень. АС находит в W имена всех вершин, которые имеют длину больше или равную длине $name(g)$ и оканчиваются на $\mu(q)$. Затем АС поочередно передает АИ метки путей в эти вершины по дереву обхода. АИ перемещается по этим путям до тех пор, пока не обнаруживает камень. АИ подбирает камень и переходит в вершину g. АС добавляет $name(q)$, $\mu(q)$, $name(g)$, $\mu(g)$ в $Al(g)$, $Ll(g)$, $Al(q)$, $Ll(q)$ соответственно. Описанные выше действия выполняются до тех пор, пока в $Ll(g)$ не окажутся все вершины из 2-окрестности вершины g.

Далее АС пытается извлечь имя вершины из списка T. Если он пуст, то все вершины графа G уже помечены и алгоритм оканчивает работу. В противном случае, АС извлекает из T имя очередной вершины, например h. По именам вершин g и h АС вычисляет метку пути из g в h по дереву обхода и передает ее АИ. АИ перемещается по этому пути в вершину h и приступает к обработке ее 2-окрестности.

С целью анализа предложенного метода, докажем следующее утверждение.

Теорема 3. Коллектив из АС и АИ восстанавливает произвольный связный неорграф G путем Д-разметки его вершин за время $O(n^3)$.

Доказательство. Покажем, что, по окончании работы алгоритма, у каждой вершины $g \in V$ в множестве $Al(g)$ содержатся все смежные g вершины графа G. Действительно, вершины из $\Gamma_g^{(2)}$, смежные g в дереве обхода, попадают в $Al(g)$ при разметке ее окрестности, а вершины, соединенные с g поперечными дереву обхода ребрами, попадают в $Al(g)$ при обработке поперечных ребер. Таким образом, по окончании работы алгоритма полностью восстанавливаются списки смежности для всех вершин графа G.

Каждая вершина графа G один раз помещается в очередь Т и один раз извлекается из нее. Таким образом, для выполнения алгоритма достаточно $O(n)$ итераций. Для разметки окрестности $\Gamma_g^{(2)}$ текущей вершины g достаточно $O(n)$ шагов. Для отыскания поперечных ребер, соединяющих вершину g с другими вершинами дерева обхода, достаточно $O(n^2)$ шагов: не более $O(n)$ шагов на отыскание поперечных ребер и не более $O(n)$ шагов на поиск по дереву обхода вершины помеченной камнем. Таким образом, временная сложность Д-разметки графа коллективом из АС и АИ ограничена $O(n^3)$.

Теорема доказана.

Выводы

В работе решена задача восстановления связного неорграфа мобильным агентом путем детерминированной разметки его вершин. Разработан алгоритм восстановления графа агентом, наблюдающим 3-окрестности посещаемых им вершин. Доказана корректность алгоритма и найдена оценка его временной сложности. В дальнейшем предполагается исследовать влияние размера наблюдаемой агентом окрестности и объема используемой им памяти на способность агента решить задачу восстановления графа.

Литература

1. Borenstein J. Navigation Mobile Robots: System and Techniques / Borenstein J., Everett B., Feng L. – A.K. Peters Ltd., Wellesley (MA), 1996. – 223 p.
2. Dudek G. Computational Principles of Mobile Robotics / Dudek G., Jenkin M. – Cambridge University Press, 2000. – 280 p.
3. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / Касьянов В.Н., Евстигнеев В.А. – СПб.: БХВ-Петребург, 2003. – 1104 с.
4. Dudek G. Map validation and Robot Self-Location in a Graph-Like World / Dudek G., Jenkin M., Milios E., Wilkes D. // Robotics and Autonomous Systems. – 1997. – Vol. 22, № 2. – P.159-178.
5. Kohavi Z. Switching and finite automata theory / Kohavi Z., Jha N. – Cambridge University Press, 2010. – 617 p.
6. Levitt T. Qualitative navigation for mobile robot / T. Levitt, D.T. Lawton // Artificial Intelligence. – 1990. – Vol. 40. – P. 306-360.
7. Кирильченко А.А. Теоретические аспекты интерпретирующей навигации мобильного робота / Кирильченко А.А., Платонов А.К., Соколов С.М. // Препринт Института прикладной математики им. М.В. Келдыша РАН. – 2002. – № 5.
8. Голубев Д.В. Об обходе графов автоматами с одной нестираемой краской. // Интеллектуальные системы. – 1999. – Т. 4. – Вып. 1-2. – С. 243-272.
9. Хопкрофт Д.Э. Введение в теорию автоматов, языков и вычислений / Хопкрофт Д.Э., Мотвани Р., Ульман Д.Д. – М. : Изд. дом «Вильямс», 2002. – 528 с.
10. Грунский И.С. Идентификация вершин помеченных графов / Грунский И.С., Сапунов С.В. // Труды ИПММ НАНУ. – 2010. – Т. 21. – С. 86-97.
11. Грунский И.С. Диагностика местоположения мобильного робота на основе топологической информации о среде / Грунский И.С., Сапунов С.В. // Искусственный интеллект. – 2011. – № 2. – С. 15-25.
12. Кормен Т. Алгоритмы: построение и анализ / Кормен Т., Лейзерсон Ч., Ривест Р. – М. : МЦНМО, 2001. – 960 с.

Literatura

1. Borenstein J. Navigation Mobile Robots: System and Techniques. A.K. Peters Ltd., Wellesley (MA), 1996. 223 p.
2. Dudek G. Computational Principles of Mobile Robotics. Cambridge University Press, 2000. 280 p.

3. Kasyanov V.N. Grafy v programmirovanii: obrabotka, vizualizacija i primenenie. SPb, BHV-Peterburg. 2003. 1104 p.
4. Dudek G. Robotics and Autonomous Systems. 1997. Vol. 22, № 2. P.159-178.
5. Kohavi Z. Switching and finite automata theory Cambridge University Press, 2010. 617 p.
6. Levitt T. Artificial Intelligence. 1990. Vol. 40. P. 306-360.
7. Kirilchenko A.A. Preprint Instituta prikladnoj matematiki im, M.V. Keldysha. RAN. 2002. № 5.
8. Golubev D.V. Intellektualnye sistemy. 1999. T. 4. Vyp. 1-2. S. 243-272.
9. Hopcroft J.E. Vvedenie v teoriju avtomatov, jazykov i vychislenij. Moscow: Izdatelskij dom “Viliams”. 2002.528 s.
10. Grunsky I.S. Transactions of IAMM NASU. 2010. Vol. 21. S.86-97.
11. Grunsky I.S. Iskusstvennyj intellect. 2011. №.2. S. 15-25.
12. Kormen T. Algoritmy: postroenie i analiz. Moscow: MCNMO. 2001. 960 p..

RESUME

I.S. Grunsky, S.V. Sapunov

Reconstruction of the Graph of Operating Environment of Mobile Robot by Vertex Labeling Sufficient for Further Navigation

The problem of robotic exploration of graph-like operating environment is considered. The problem is: an unknown environment modeled as finite simple connected undirected vertex labeled graph is given, formulate an exploration strategy for the robot, so that, after carrying out the actions specified by the strategy, the robot will have formed a representation of its environment sufficient for solving navigation tasks i.e. a map.

The authors call vertex labeled graph deterministic if all vertices in neighborhood of each vertex have distinct labels. The sufficiency of deterministic graphs for solving navigation tasks is demonstrated.

The authors propose polynomial time exploration algorithm for collective of robot-explorer and robot-supervisor. Robot-explorer traverses the graph, observes the local neighborhoods of the vertices and changes the environment by setting labels at the vertices in deterministic manner. Robot-explorer sends information about vertex labeling to robot-supervisor. Robot-supervisor forms a map of the environment based on this information. To solve the problem of determining when the robot-explorer has returned to a previously visited vertex during map building, the authors use portable marker as a part of exploration strategy.

Статья поступила в редакцию 15.06.2012.

.